Week 3

We have already modelled our mobile robot as a system i.e., the unicycle model. This week we are going to design a completely new system. The reason for this is, as accurate and precise the unicycle model was, it is not very generalized. For instance, you want to design a controller for a drone you will have to design a model from scratch for that. For this purpose, we are going to come up with a new system that is more general and more representative of a variety of different models. Then we can implement that controller to any robot, be it a 2-wheel differential drive or even a drone by simply transforming to the specific model.

State Space Systems

In order to derive a more general model, let us assume our robot to be a point. Let us also assume that the robot can only move forward or backward on a line. So, it would be a 1-Dimensional point, a point that can move on a line. Let us model this system. If p represents the position of the point on the robot then, the acceleration of the robot is given by

$$u = \ddot{p} \tag{1}$$

To represent this simple equation in a more compact form, let us take two variables $x_1$and x_2. Let,

$$x_1 = p \quad \rightarrow \quad \dot{x}_1 = x_2, \tag{2}$$
$$x_2 = \dot{p} \quad \rightarrow \quad \dot{x}_2 = u \tag{3}$$

Let us store these variables in a vector x,

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Then,

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = p = x_1 = [1 \;\; 0]x$$

Or,

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Where,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \;\; 0]$$

Are known as the state space equations.

If we try to do this for a point that can move in 2 dimensions, i.e., in both x and y directions let us see what happens. This time let $p_x \; and \; p_y$, give the positions in x and y axes respectively. Then the accelerations would be given by,

$$\ddot{p}_x = u_x$$
$$\ddot{p}_y = u_y \, ,$$

Now since we have to keep track of two positions, we will need 4 variables (we needed only two in the last case). Let,

$$x_1 = p_x$$

$$x_2 = \dot{p}_x$$

$$x_3 = p_y$$

$$x_4 = \dot{p}_y$$

Surprisingly the state space equation is also applicable here.

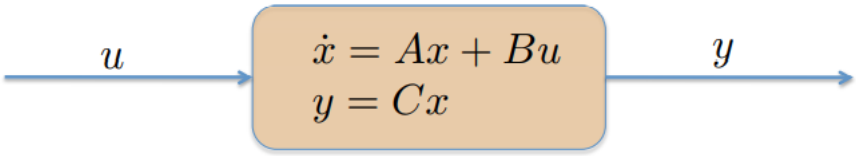
$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Where,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

here $y \; and \; u$ will also be vectors representing $p_x, p_y \; and \; u_x, u_y$ respectively. Can you see the pattern here? Most systems can be simply modelled by the state space equation with only the matrices $A, B \; and \; C$ changing. This is essentially the state-space form of a Linear Time Invariant system.

$x - state$
$y - output$
$u - input$



So given any system, we can measure the output y and accordingly vary input u to get desired output y. Isn't that great? So, we can just convert our unicycle model into a state space form too right. Let's try just that. The position of the robot in a unicycle model is given by $x, y \; and \; \emptyset$. If you recall the equations that describe the unicycle model are,

$$x = v\cos\emptyset$$
$$\dot{y} = v\sin\emptyset$$
$$\dot{\emptyset} = \omega$$

The problem with these equations is that, they are not linear. And the State-Space form we have derived is applicable only to **Linear** Time Invariant systems. In this vast universe most objects are not linear in nature. Very few systems are linear. But this state-space form is very useful. Now we need a way to fit a non-linear system into the state-space form meant for LTI systems. To do this we need to linearize the system.

Linearization

Given a non-linear model,

$$\dot{x} = f(x,u), \qquad y = h(x)$$

Where the functions define the system's dynamics. We can linearize this model by finding a "local" linear model around an operating point at which the system behaves in a linear fashion. For instance let us say a system is linear when we approximate it's equations around a certain point (x_o, y_o)

$$(x_o, y_o) \rightarrow (x = x_o + \delta x, \qquad u = u_0 + \delta u)$$

Then the new equation describing the system becomes,

$$\dot{\delta}x = \dot{x} - \dot{x}_o = \dot{x} = f(x_o + \delta x, u_o + \delta u)$$

Taylor expansion of $\dot{\delta}x = f(x_o + \delta x, u_o + \delta u)$ is given by,

$$\dot{\delta}x = f(x_o, u_o) + \frac{\partial f}{\partial x}(x_o, u_0)\delta x + \frac{\partial f}{\partial u}(x_o, u_0)\delta u + H.O.T$$
$$y = h(x_o + \delta x) = h(x_o) + \frac{\partial h}{\partial x}(x_o, u_0)\delta x + H.O.T$$

The values of the matrices for the state space form can be obtained from the above expansion.

$$A = \frac{\partial f}{\partial x}(x_o, u_0)$$
$$B = \frac{\partial f}{\partial u}(x_o, u_0)$$
$$C = \frac{\partial h}{\partial x}(x_o, u_0)$$

The above expressions are obtained under the assumption $f(x_o, u_o) = h(x_o) = 0$. That is we assume that the functions become 0 at the operating point.

If the function f and h are given by the vectors,

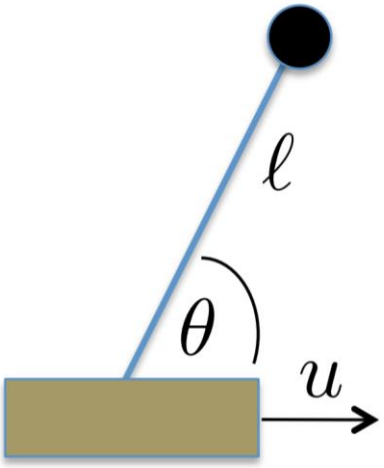
$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \qquad h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix}$$

Then the jacobians are given as,

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}, \quad \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}$$

The value of the above jacobians at the operating point give the state matrices.

Let us solve an example to understand how to solve those equation better. Consider an inverted pendulum.



This system is described by the following equation

$$\ddot{\theta} = \frac{g}{l}\sin\theta + u\cos\theta$$

Let us take consider two variables x_1 and x_2 such that

$$x_1 = \theta,$$
$$x_2 = \dot{\theta}$$

Then the output is given by

$$y = x_1$$

The functions describing the system are given by the vectors,

$$f(x,u) = \begin{bmatrix} x_2 \\ \frac{g}{l}\sin x_1 + u\cos x_1 \end{bmatrix}$$

$$h(x) = x_1$$

Let us take our optimal point to be

$$(x_o, u_o) = (0,0)$$

Then the matrices for the state space representation are derived as

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l}\cos x_1 & 0 \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 \\ \cos x_1 \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} \end{bmatrix}_{(0,0)} = [1 \quad 0]$$

Now even a non-linear system such as an inverted pendulum can be represented as a state space system.

Task:

1. Now try to find the state space matrices for the unicycle model.
2. Simulate the pioneer robot on coppeliasim to reach a goal position using PID control. The goal position is given by a dummy object in coppeliasim. (Hint: The coordinates of any object can be accessed using a function simxGetObjectPosition).

References

1. (1463) Equations of Motion for the Inverted Pendulum (2DOF) Using Lagrange's Equations - YouTube
2. Derivation of Equations of Motion for Inverted Pendulum Problem (mcmaster.ca)